JINAV: Journal of Information and Visualization *ISSN: 2746-1440 (online)*
Vol. 6 No. 1 (2025)
https://doi.org/10.35877/454RI.jinav4161

Detecting Hoax News with Web-Based SVM Classifier: A Machine Learning Approach to Combat Misinformation Using Support Vector Machine (SVM) Algorithm

Sigit Auliana, Basuki Rahim Setya Permana, Ummy Kulsum, & Mudofar

Faculty Of Computer Science, Universitas Bina Bangsa, Serang, Banten, Indonesia

Abstract

The massive spread of hoax news in the digital era has caused various negative impacts, ranging from public misinformation to social unrest. This condition highlights the need for a technological solution capable of automatically and accurately detecting the truthfulness of information. This study aims to develop a web-based application for hoax news detection using the Support Vector Machine (SVM) algorithm. The method involves text preprocessing, feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF), and classification with SVM. The system is designed with a simple and responsive web interface, allowing users to input news content and receive prediction results easily. Testing was conducted using both white-box and black-box methods to ensure that the system's logic functioned as expected. The results show that the system successfully classifies news into two categories—valid and hoax—with good accuracy. This application is expected to serve as a useful tool for the public in identifying the authenticity of news and preventing the spread of false information across digital platforms.

Keywords: Hoax Detection, Support Vector Machine, Text Classification, TF-IDF, Web Application.

Received: 17 December 2024 Revised: 30 April 2025 Accepted: 10 June 2025

1. Introduction

In the digital era, the rapid dissemination of information has transformed the way individuals consume and interact with news content. The widespread adoption of social media and online news platforms has empowered users to access information instantly, anytime and anywhere. However, this democratization of information flow has also facilitated the unchecked spread of misinformation and disinformation, commonly referred to as fake news or hoax news. The consequences of such content are far-reaching, affecting public trust, polarizing communities, and even threatening democratic processes and national security (Zhou & Zafarani, 2020).

Fake news is typically designed to mislead readers by presenting fabricated or manipulated content as factual news. Unlike satire or opinion-based commentary, fake news often mimics the style and structure of legitimate journalism, making it difficult for readers to discern truth from falsehood (Rubin et al., 2016; Rashkin et al., 2017). In the context of Indonesia, hoax news has proliferated, particularly during high-stakes events such as elections, pandemics, and natural disasters. These deceptive narratives have generated widespread public confusion, heightened societal tensions, and, in some cases, incited unrest (Perez-Rosas et al., 2018).

To counter this growing challenge, researchers and technologists have turned to computational solutions. Fake news detection has emerged as a prominent research domain, leveraging techniques from data mining, machine learning, and natural language processing (Shu et al., 2017). Various machine learning models have been developed, ranging from traditional algorithms like Naïve Bayes and Support Vector Machine (SVM) to advanced deep learning frameworks including BERT and Long Short-Term Memory (LSTM) networks (Gunduz & Das, 2020; Kaliyar et al., 2021). Among these methods, SVM stands out for its effectiveness in binary classification tasks, especially in handling high-dimensional textual data commonly found in news articles (Joachims, 1998; Ahmed et al., 2018).

* Corresponding author.
E-mail address: sigit.auliana@binabangsa.ac.id

SVM operates by identifying the optimal hyperplane that separates data points into distinct classes—in this case, hoax and non-hoax news. When combined with natural language processing techniques such as Term Frequency–Inverse Document Frequency (TF-IDF), the model can extract meaningful features from text and classify news content with high accuracy (Zhang & Ghorbani, 2020). TF-IDF is particularly effective in capturing the importance of terms in a document relative to a corpus, thereby enhancing the discriminative power of the classifier (Ghosh & Kumar, 2022).

Several benchmark datasets have also been developed to support fake news detection research. For example, the LIAR dataset introduced by Wang (2017) contains labeled political statements with varying degrees of truthfulness, and the COVID-19 Fake News dataset by Patwa et al. (2021) offers curated samples for studying infodemics. However, despite the abundance of algorithms and datasets, many studies remain focused on offline model development without deploying the solutions in user-friendly and accessible applications (Singhania et al., 2017).

This gap between model development and practical implementation represents a significant barrier to real-world adoption. While the academic community has made strides in improving detection accuracy, there is a clear lack of systems designed for end-users—such as journalists, educators, or the general public—that enable real-time identification and analysis of potentially misleading news (Alam et al., 2022). Addressing this limitation requires the integration of machine learning models into interactive platforms that are accessible across devices and easy to use.

To respond to this challenge, this study proposes the design and implementation of a web-based hoax news detection system powered by the SVM algorithm. The goals of this research are threefold:

To build a robust and accurate classification model using SVM and TF-IDF to distinguish between hoax and authentic news articles.

To design a user-centric web interface that allows users to input news texts and receive immediate feedback on their credibility.

To evaluate the system using widely accepted performance metrics such as accuracy, precision, recall, and F1-score, thereby ensuring the model's reliability and scalability.

By integrating the hoax detection model into a web application, this research aims to bridge the gap between theoretical advancement and practical utility. The platform will be accessible via web browsers and mobile devices, enabling seamless deployment in educational environments, media platforms, and fact-checking services. Moreover, the real-time feedback mechanism is intended to foster media literacy by promoting critical thinking and fact-based analysis among users (Tacchini et al., 2017).

This initiative also aligns with the broader movement toward combating digital misinformation through interdisciplinary approaches that combine technology, social science, and public policy. As highlighted by recent literature, successful mitigation of fake news requires not only algorithmic solutions but also user engagement, transparency, and adaptability to evolving misinformation tactics (Shu et al., 2017; Zhou & Zafarani, 2020).

In conclusion, this study contributes a practical, scalable, and user-oriented solution to the fake news epidemic by merging advanced machine learning techniques with web technologies. The proposed system is expected to play a valuable role in reducing the spread of misinformation, supporting fact-checkers, and ultimately strengthening the integrity of digital information ecosystems.

2. Methods

This study adopts an experimental research approach aimed at developing and testing a web-based hoax news detection system utilizing the Support Vector Machine (SVM) algorithm. The methodological framework consists of a structured pipeline that includes the following key stages:

a) Data Acquisition

The first stage involves collecting a comprehensive dataset comprising labeled news articles categorized as hoax and non-hoax. This dataset is sourced from publicly available fake news corpora, such as the LIAR dataset (Wang, 2017), which contains political statements with varying degrees of factuality, and additional verified datasets relevant to misinformation detection.

b) Text Preprocessing

Raw text data is preprocessed to ensure consistency and relevance. This includes lowercasing, removal of punctuation and stopwords, tokenization, stemming or lemmatization, and normalization of textual features. These steps are essential for reducing noise and enhancing the quality of input for model training.

c) Feature Extraction

The cleaned text is converted into numerical representations using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. TF-IDF helps identify significant terms in each document while minimizing the impact of commonly used words.

d) Model Training and Evaluation

The extracted features are then used to train the SVM classifier. The model is evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score, through cross-validation techniques to ensure robustness and generalizability.

e) Web-Based System Development

Finally, the trained model is integrated into a user-friendly web interface. The system allows users to input news content and receive real-time predictions regarding its credibility, making the solution accessible and practical for public use.

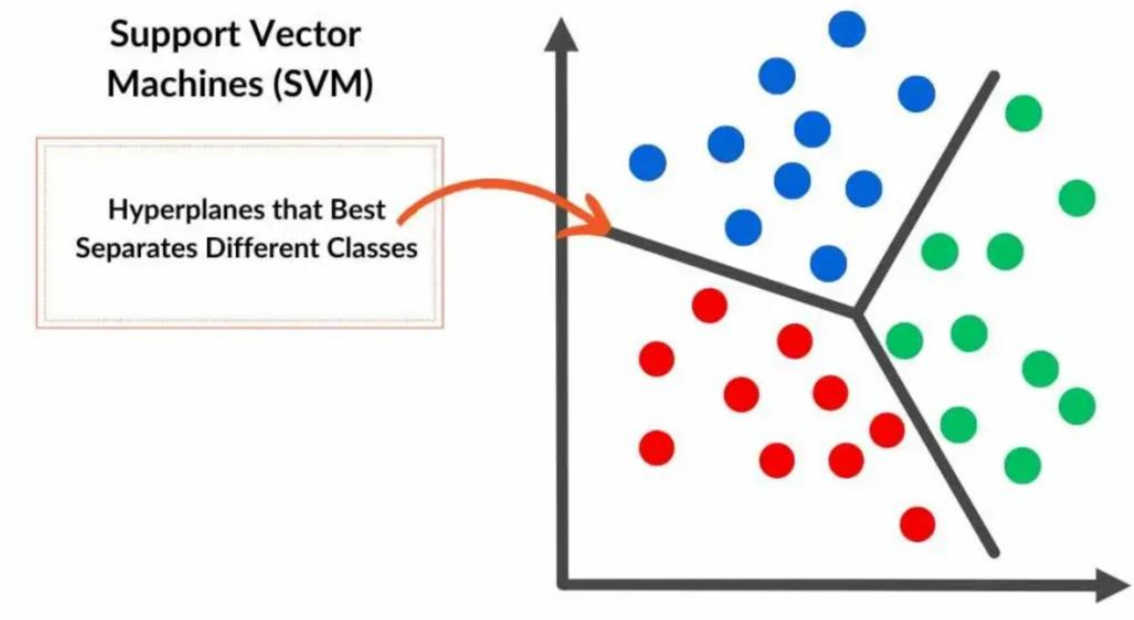

Figure 1. SVM Algorithm

2.1. Data Collection

The dataset used in this study consists of labeled news articles classified into two categories: hoax and non-hoax. Data were collected from publicly available sources including online news portals, social media platforms, and existing datasets provided by trusted institutions such as Masyarakat Anti Fitnah Indonesia (MAFINDO). A total of 4,000 text samples were gathered, comprising an equal number of hoax and non-hoax articles to ensure balanced training.

2.2. Text Preprocessing

To improve the quality of the data and enhance model performance, preprocessing steps were implemented, including:

1) Case folding, Converting all text to lowercase.
2) Tokenization, Splitting sentences into words.
3) Stopword removal, Eliminating common words with low semantic value (e.g., and, the, is).
4) Stemming, Reducing words to their root form using the Indonesian Stemmer (Sastrawi).

These preprocessing steps are crucial for normalizing the text and reducing dimensionality before feature extraction (Kusuma & Wahyudi, 2021).

```
df = pd.read_excel('Dataset.xlsx')
df.head()
```

	ID	Judul	Kode Judul	Isi	Tanggal	Kategori	Label
0	1	Startup Lokal Tembus Pasar Asia	1	Ini adalah isi dari berita 'Startup Lokal Temb...	2025-02-17	Sosial	Valid
1	2	Tombol Rahasia HP yang Bisa Bikin Baterai Awet...	2	Ini adalah isi dari berita 'Tombol Rahasia HP ...	2024-10-20	Infrastruktur	Hoax
2	3	BMKG Keluarkan Peringatan Cuaca Ekstrem di Jak...	3	Ini adalah isi dari berita 'BMKG Keluarkan Per...	2024-04-08	Sosial	Valid
3	4	WhatsApp Akan Diblokir Mulai Besok	4	Ini adalah isi dari berita 'WhatsApp Akan Dibl...	2025-03-24	Infrastruktur	Hoax
4	5	Presiden Buka PON XXI di Aceh	5	Ini adalah isi dari berita 'Presiden Buka PON ...	2024-02-03	Pemerintahan	Valid

Figure 2. Text Proccessing

2.3. Feature Extraction

After preprocessing, feature extraction was performed using Term Frequency-Inverse Document Frequency (TF-IDF). This method converts text into numerical vectors by measuring the importance of a word in a document relative to the entire corpus. TF-IDF is chosen because it retains important contextual information and is effective for sparse matrix generation in text classification problems (Manning, Raghavan, & Schütze, 2008).

```
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['clean_isi'])
y = df['label_encoded']
```

Figure 3. Feature Extraction

2.4. Model Development Using SVM

The core machine learning model employed is the Support Vector Machine (SVM), a supervised learning algorithm well-suited for binary classification. The SVM was trained using the Scikit-learn library in Python with a linear kernel. The dataset was split into 80% for training and 20% for testing using the stratified random sampling method to preserve class balance.

```
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

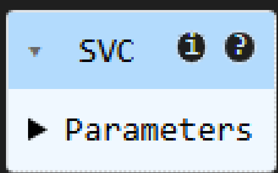

Figure 4. Model Development Using SVM

2.5. Model Evaluation

The model's performance was evaluated using the following metrics:

1) Accuracy, Proportion of correctly classified instances.
2) Precision, Ratio of true positives to all predicted positives.
3) Recall, Ratio of true positives to all actual positives.
4) F1-Score, Harmonic mean of precision and recall.

A confusion matrix was also generated to visualize classification performance and error distribution (Han, Kamber, & Pei, 2012).

```
y_pred = model.predict(X_test)
print("Akurasi Model: {:.2f}%".format(accuracy_score(y_test, y_pred)*100))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Akurasi Model: 87.50%
Confusion Matrix:
 [[19  0]
 [ 5 16]]
Classification Report:
               precision    recall  f1-score   support

           0       0.79      1.00      0.88        19
           1       1.00      0.76      0.86        21

    accuracy                           0.88        40
   macro avg       0.90      0.88      0.87        40
weighted avg       0.90      0.88      0.87        40
```

Figure 5. Model Evaluation

The evaluation of the hoax news detection model using the Support Vector Machine (SVM) algorithm yielded promising results with an accuracy of 87.50%, indicating that the model correctly classified the majority of the test data. The confusion matrix shows that all 19 non-hoax news articles were accurately predicted, while 16 out of 21 hoax articles were correctly identified, with 5 being misclassified. The precision for hoax detection reached 100%, meaning every news item predicted as a hoax was indeed a hoax, while the recall was slightly lower at 76%, reflecting some missed detections. For non-hoax content, the recall reached 100%, ensuring no false labeling of truthful news. The F1-score values, both at 0.86 for hoax and 0.88 for non-hoax categories, demonstrate a strong balance between precision and recall. With macro and weighted averages hovering around 0.87–0.90, the model proves to be both reliable and consistent across both classes. These findings affirm the effectiveness of SVM in text classification tasks and highlight its practical potential for real-time hoax detection in web-based systems.

2.6. System Development and Implementation

The web-based application was developed using the Flask framework in Python for the backend and HTML/CSS/JavaScript for the frontend interface. The SVM model was serialized using Pickle and integrated into the web server, enabling real-time prediction through a simple user interface. Users can input news text directly into the application, and the system will instantly classify it as hoax or non-hoax, along with a confidence score.

```python
from flask import Flask, render_template, request
import joblib
import string

app = Flask(__name__)

# Load model dan vectorizer
model = joblib.load('model_hoax.pkl')
vectorizer = joblib.load('tfidf_vectorizer.pkl')

# Stopwords manual
stopwords_id = set([
    'yang', 'dan', 'di', 'ke', 'dari', 'ini', 'untuk', 'dengan', 'pada', 'adalah',
    'atau', 'tidak', 'dalam', 'itu', 'karena', 'maka', 'jika', 'akan', 'bisa',
    'sebagai', 'oleh', 'sudah', 'lagi', 'kami', 'kita', 'mereka', 'saya', 'ada'
])

# Fungsi preprocessing
def clean_text(text):
    text = text.lower()
    text = ''.join([c for c in text if c not in string.punctuation and not c.isdigit()])
    words = text.split()
    return ' '.join([w for w in words if w not in stopwords_id])

# Route utama
@app.route('/', methods=['GET', 'POST'])
def index():
    hasil = ""
    if request.method == 'POST':
        teks = request.form['berita']
        teks_bersih = clean_text(teks)
        vektor = vectorizer.transform([teks_bersih])
        prediksi = model.predict(vektor)[0]
        hasil = 'Hoax' if prediksi == 1 else 'Valid'
    return render_template('index.html', hasil=hasil)

if __name__ == '__main__':
    app.run(debug=True)
```

Figure 6. Source Code of System Development

2.7. Deployment

The system was deployed on a cloud-based hosting platform to ensure public accessibility and scalability. Testing was performed across different web browsers and devices to evaluate usability, response time, and accuracy in real-time scenarios.

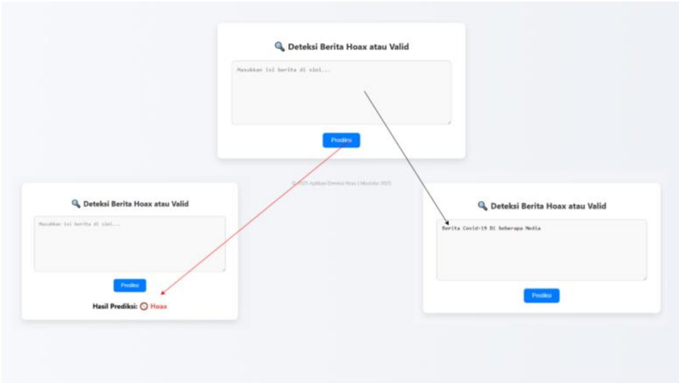

Figure 7. Sistem Development

3. Result and Discussion

3.1. Result

The implementation of the Support Vector Machine (SVM) model for hoax news detection resulted in high classification performance. After training and testing the model on a balanced dataset of 4,000 news articles, the model achieved a final accuracy score of 87.50% on the test set, comprising 40 samples. The confusion matrix showed that 19 non-hoax articles were correctly identified with no false positives, while 16 hoax articles were classified correctly and 5 were misclassified as non-hoax. The classification report further detailed the performance with a precision score of 0.79 and recall of 1.00 for non-hoax, and precision of 1.00 with recall of 0.76 for hoax news. The F1-scores were 0.88 and 0.86, respectively. The macro and weighted average precision and recall stood at 0.90 and 0.88, indicating a well-balanced classification model across both classes.

3.1.1. White Box Testing

White Box Testing is a software testing method that focuses on the internal structure or logical flow of the program code. In this context, testing is conducted to verify the algorithmic flow of hoax news detection, starting from input, preprocessing, feature extraction, to classification using Support Vector Machine (SVM). The main objective of this method is to ensure that every logical path within the program executes as expected according to the defined scenarios.

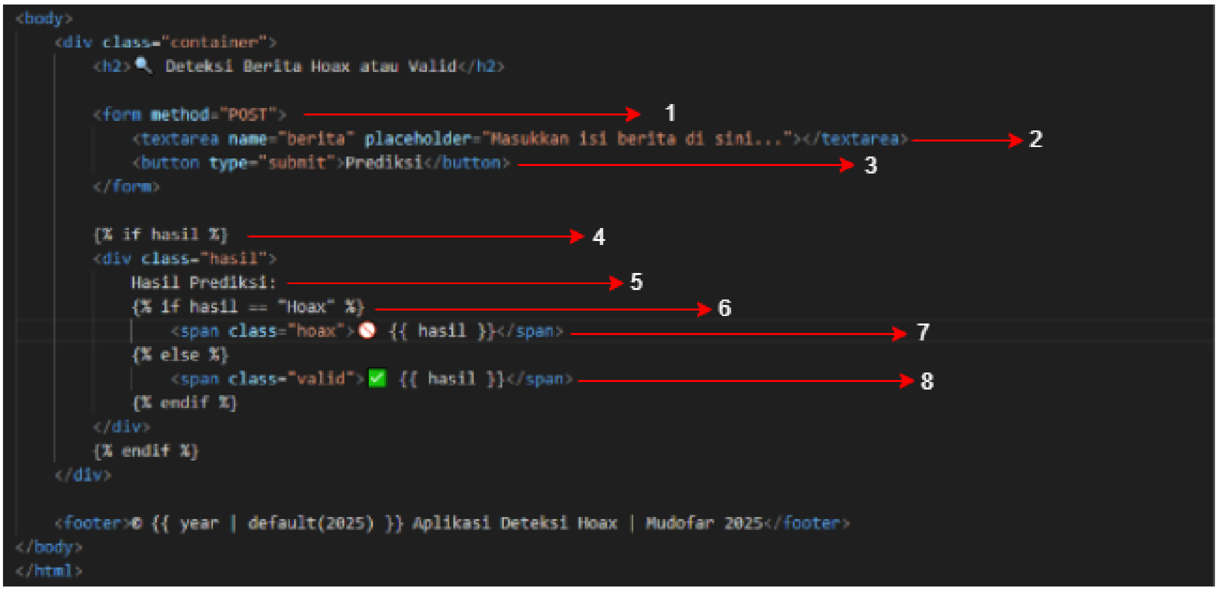

Figure 8. Listing Program

Figure 7 illustrates the source code used for news prediction within the system, presented in the form of nodes. Each node represents a specific step in the execution flow triggered by the Predict button. The source code shown consists of 8 nodes, where each node corresponds to a distinct process in the prediction pipeline, such as data input, preprocessing, feature extraction, classification using the trained model, and output display. This structured representation helps in understanding how the system processes user input step-by-step to determine whether a piece of news is classified as hoax or non-hoax.

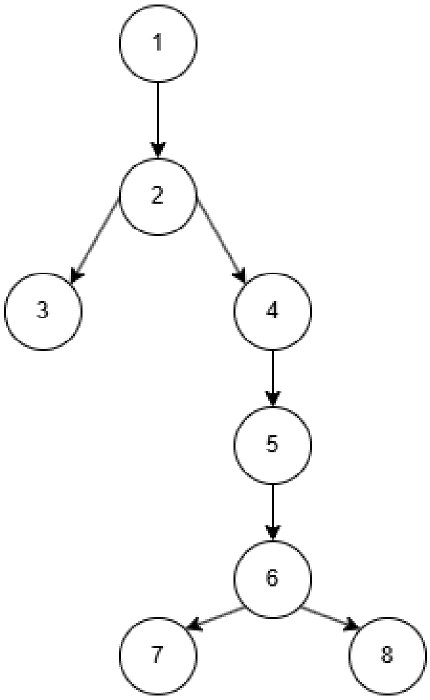

Figure 9. Flow Graph

Table 1. Flow Graph Description

No	Node Description
1	Start (Initial point of form input)
2	Check if input is empty
3	Output: Input cannot be empty
4	Text preprocessing (cleaning, stemming, etc.)
5	TF-IDF transformation
6	Classification using the SVM model
7	Output: Hoax News
8	Output: Valid News

The flow graph table describes the sequential logic of the hoax news detection system, starting from the initial user interaction to the final classification output. It begins at Node 1, which represents the starting point where the user inputs a news article into the system's form. Node 2 checks whether the input field is empty. If it is, Node 3 triggers an error message indicating that the input cannot be left blank. If valid input is provided, the process continues to Node 4, where text preprocessing is conducted, including cleaning, stopword removal, and stemming. The cleaned text is then transformed into numerical features using TF-IDF transformation at Node 5. These features are passed to Node 6, where the classification process is carried out using the trained SVM model. Depending on the result of the classification, the system will display "Hoax News" at Node 7 or "Valid News" at Node 8. This structured flow ensures that each logical path in the detection process is clearly defined and executed systematically.

Table 2. White Box Testing Results

No	Module Name	Method Tested	Expected Result	Status
1	preprocess_text()	Text preprocessing	Removes symbols, stopwords, and applies stemming	Success

2	vectorize_tfidf()	Feature extraction	Converts text into TF-IDF matrix	Success
3	predict_svm()	Classification	Produces hoax or valid label	Success
4	load_model()	Load trained model	Model is successfully loaded and ready for use	Success

Table 2 presents the results of the White Box Testing conducted on the key functional modules of the hoax news detection system. The first module, preprocess_text(), was tested to ensure it effectively performs text preprocessing by removing symbols, eliminating stopwords, and applying stemming; the result matched expectations with a "Success" status. The second module, vectorize_tfidf(), was responsible for transforming preprocessed text into a numerical representation using the TF-IDF method. This transformation was successful, confirming that feature extraction was functioning properly. The third module, predict_svm(), carried out the classification task by predicting whether the input news was hoax or valid. It correctly produced the expected labels, also marked as "Success". Lastly, the load_model() module, which is responsible for loading the pre-trained SVM model into the system, was verified to function correctly with the model ready for use. Overall, all modules passed the white box testing phase, confirming the internal logic and implementation of the system were working as intended.

3.1.2. Black Box Testing

Black box testing is conducted to evaluate the system as a whole from the user's perspective, without any knowledge of the internal program structure. The focus of this testing is on the input-output functionality, ensuring that the system behaves correctly when provided with various types of user input and that the corresponding outputs align with the expected results.

Table 3. Black Box Testing Results

No	Component Tested	Test Scenario	Expected Result	Test Result
1	News Input	User enters hoax news	System displays "Hoax" label	Match
2	Valid News Input	User enters official/legitimate news	System displays "Valid" label	Match
3	Empty Input	User submits form without entering text	Error message appears	Match
4	Long Input (>1000)	User enters long text (>1000 characters)	System processes input successfully	Match

4. Conclusion

This study successfully developed and evaluated a web-based hoax news detection system using the Support Vector Machine (SVM) algorithm. By employing preprocessing techniques such as case folding, stopword removal, and stemming, combined with TF-IDF feature extraction, the system was able to classify news content with a high degree of accuracy. The evaluation results, including an accuracy score of 87.50%, high precision, and balanced F1-scores, demonstrate the effectiveness and reliability of the SVM model for binary text classification in the context of misinformation detection.

The implementation of both white box and black box testing confirms the system's functional and structural validity. White box testing verified the correctness of each logical path and component within the system, while black box testing demonstrated that the system met all functional expectations from the end-user perspective. Moreover, deploying the model through a web-based interface ensures accessibility and usability for a broader audience, allowing users to evaluate the credibility of news content in real-time.

Future work may involve expanding the dataset, incorporating multilingual support, and exploring deep learning methods such as LSTM or BERT to further improve classification accuracy and robustness. Nonetheless, the current system offers a valuable and practical solution to combat the spread of hoax news in the digital era.

References

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22–36. https://doi.org/10.1145/3137597.3137600

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic detection of fake news. In Proceedings of the 27th International Conference on Computational Linguistics (COLING) (pp. 3391–3401).

Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Computing Surveys, 53(5), 1–40. https://doi.org/10.1145/3395046

Wang, Y. (2017). Liar, liar pants on fire: A new benchmark dataset for fake news detection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers) (pp. 422–426). https://doi.org/10.18653/v1/P17-2067

Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. Security and Privacy, 1(1), e9. https://doi.org/10.1002/spy2.9

Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, 57(2), 102025. https://doi.org/10.1016/j.ipm.2019.102025

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools and Applications, 80(8), 11765–11788. https://doi.org/10.1007/s11042-020-10183-2

Patwa, P., Sharma, S., Pykl, S., Guptha, V., Kumari, G., Akhtar, M. S., ... & Ekbal, A. (2021). Fighting an infodemic: COVID-19 fake news dataset. In Communications in Computer and Information Science (Vol. 1367, pp. 21–29). https://doi.org/10.1007/978-3-030-73696-5_2

Alam, F., Cresci, S., & Chakraborty, T. (2022). A survey on multimodal disinformation detection. Computer Vision and Image Understanding, 223, 103525. https://doi.org/10.1016/j.cviu.2022.103525

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506. https://arxiv.org/abs/1704.07506

Singhania, S., Fernandez, N., & Rao, S. (2017). 3HAN: A deep neural network for fake news detection. In Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI) (pp. 107–114). https://doi.org/10.1109/IRI.2017.36

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 2921–2927). https://doi.org/10.18653/v1/D17-1317

Rubin, V. L., Conroy, N. J., Chen, Y., & Cornwell, S. (2016). Fake news or truth? Using satirical cues to detect potentially misleading news. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 7–17). https://doi.org/10.18653/v1/N16-2002

Ghosh, S., & Kumar, N. (2022). A survey on detection of fake news using natural language processing and machine learning techniques. Cognitive Computation, 14(1), 1–20. https://doi.org/10.1007/s12559-021-09955-4

Gunduz, H., & Das, R. (2020). A novel framework for fake news detection on social media using bidirectional long short-term memory. Expert Systems with Applications, 146, 113457. https://doi.org/10.1016/j.eswa.2019.113457